

---

# **doto Documentation**

*Release 0.2*

**Benjamin Zaitlen**

March 30, 2014



<b>1</b>	<b>Installing</b>	<b>3</b>
<b>2</b>	<b>Getting Started</b>	<b>5</b>
<b>3</b>	<b>Currently Supported Services</b>	<b>7</b>
3.1	Droplets . . . . .	7
3.2	Images . . . . .	7
3.3	Domains . . . . .	8
3.4	Management . . . . .	9
3.5	Configuration . . . . .	9
3.6	Public Images . . . . .	9
3.7	Requirements . . . . .	10
3.8	Release Notes . . . . .	10
<b>4</b>	<b>Indices and tables</b>	<b>11</b>



DOTO is an open-source Python 2/3 interface to [Digital Ocean's API](#).  
Source code located on [Github](#).



---

## Installing

---

You can install `doto` with `conda` or `pip`:

```
$ pip install doto
$ conda install doto
```

or install [from source](#)

```
$ git clone https://github.com/quasiben/doto.git
$ cd doto
$ python setup.py install
```

Why yet another Python library for Digital Ocean?

- Logging
- boto like interface for integrated development
- Optionally formatted tables for inline exploration
- CLI





---

## Getting Started

---

To get started with doto create a **.dotorc** file in a directory named, **~/doto**, with your **api\_key** and **client\_id** listed:

```
[Credentials]
client_id = XXXXXXXXXXXXXXXX
api_key = 99999999999999999999999999999999
```

You are now ready to use doto:

```
import doto
d0 = doto.connect_d0()

new_key = d0.create_key_pair('my_new_key_pair')
droplet = d0.create_droplet(name='Random',
                             size_id=66, #512MB
                             image_id=1341147, #Docker 0.7 Ubuntu 13.04 x64
                             region_id=1, #New York
                             ssh_key_ids=[new_key['id']]
                             )
```

Or use doto directly from the command line:

```
$ doto --help
$ doto start --name Random --size_id 66 --image_id 2158507 --region_id 1 --ssh_key_ids 89221
$ doto listdroplets
$ ...
```

Doto is designed to support both Python 2.7 and Python 3. A number of functions add a bit more than just returning json converted dicts. For example, **Images** and **Droplets** are objects within doto where data such as, event\_id, ip\_address, status, etc. are stored as attributes with respect to the individual object:

```
In [1]: import doto
d0
In [2]: d0 = doto.connect_d0()

In [22]: images = d0.get_all_images()
>>> Getting /images

In [23]: images
Out[23]:
[Image:490208,
 Image:568111,
 Image:633923,
 Image:714697,
 Image:1420886,
```



---

## Currently Supported Services

---

### 3.1 Droplets

Droplets are the atomic unit of compute instances on the Digital Ocean cloud service. They are available in a variety of RAM, HD, CPU configurations.

#### 3.1.1 Creating Droplets

```
droplet = d0.create_droplet(name='Random',
                             size_id=66, #512MB
                             image_id=1341147, #Docker 0.7 Ubuntu 13.04 x64
                             region_id=1, #New York
                             ssh_key_ids=[new_key['id']]
                             )
```

#### 3.1.2 Getting a specific droplet

```
droplet = d0.get_droplet(923125)
```

#### 3.1.3 Getting all droplets

```
droplets = d0.get_all_droplets()
```

#### 3.1.4 Droplet API

### 3.2 Images

A droplet is created using a pre-defined **Image**. Users can select from nearly 40 *Public Images* or from snapshots and backups previously created.

### 3.2.1 Creating Snapshots

```
>>>d0.create_snapshot(name='My New Snapshot')
```

### 3.2.2 Getting a specific image

```
>>>images = d0.get_image(1860934)
```

### 3.2.3 Getting all images

```
>>>images = d0.get_all_images()
```

```
>>>images
[Image:490208,
 Image:568111,
 Image:633923,
 Image:714697,
 Image:1420886,
 Image:1898676,
 Image:2003826, ...]
```

```
>>>d0.get_all_images(table=True)
```

```
| Ubuntu      | None | True | 1505699 | Ubuntu 13.10 x64 |
| Ubuntu      | None | True | 1608711 | Ruby on Rails on Ubuntu 12.10 (Nginx + Unicorn) |
| CentOS      | None | True | 1646467 | CentOS 6.5 x64 |
| CentOS      | None | True | 1646732 | CentOS 6.5 x32 |
| Ubuntu      | None | True | 1687372 | Redmine on Ubuntu 12.04 |
| Ubuntu      | None | True | 1860934 | Ghost 0.4.0 on Ubuntu 12.04 |
| Ubuntu      | None | True | 2105243 | GitLab 6.5.1 CE |
| Ubuntu      | None | True | 2118237 | Dokku-v0.2.1 on Ubuntu 13.04 |
| Ubuntu      | None | True | 2158507 | Docker 0.8 Ubuntu 13.04 x64 |
...

```

### 3.2.4 Image API

## 3.3 Domains

Digital Ocean allows users to easily setup and control hostnames and subdomains for existing droplets. Simply point the DNS of your host provider to:

- ns1.digitalocean.com
- ns2.digitalocean.com
- ns3.digitalocean.com

For full documentation on setup please read: [How To Set Up a Host Name with DigitalOcean](#).

### 3.3.1 Creating Domains

```
domain = d0.create_domain(name='myurl.com', ip_addr='555.55.5.55')

#or with a droplet
droplet = d0.create_droplet(..)
domain = d0.create_domain(name='myurl.com', ip_addr=droplet.ip_address)
```

### 3.3.2 Getting a specific domain

```
droplet = d0.get_domain(domain_id=555555)
```

### 3.3.3 Getting all droplets

```
domains = d0.get_all_domains()
```

### 3.3.4 Droplet API

## 3.4 Management

### 3.4.1 DigitalOcean API

## 3.5 Configuration

An example `~/ .doto/ .dotorc` file should look like:

```
[Credentials]
client_id = XXXXXXXXXXXXXXXX
api_key = 99999999999999999999999999999999
```

## 3.6 Public Images

distribution	id	name
Arch Linux	361740	Arch Linux 2013.05 x32
Arch Linux	350424	Arch Linux 2013.05 x64
CentOS	1601	CentOS 5.8 x64
CentOS	562354	CentOS 6.4 x64
CentOS	1602	CentOS 5.8 x32
CentOS	1646467	CentOS 6.5 x64
CentOS	1646732	CentOS 6.5 x32
CentOS	376568	CentOS 6.4 x32
Debian	12573	Debian 6.0 x64
Debian	303619	Debian 7.0 x32
Debian	308287	Debian 7.0 x64
Debian	12575	Debian 6.0 x32
Fedora	32399	Fedora 17 x32 Desktop
Fedora	32419	Fedora 17 x64 Desktop

Continued on next page

Table 3.1 – continued from previous page

distribution	id	name
Fedora	32428	Fedora 17 x64
Fedora	697056	Fedora 19 x32
Fedora	696598	Fedora 19 x64
Fedora	32387	Fedora 17 x32
Ubuntu	2105243	GitLab 6.5.1 CE
Ubuntu	2118237	Dokku-v0.2.1 on Ubuntu 13.04
Ubuntu	2158507	Docker 0.8 Ubuntu 13.04 x64
Ubuntu	14097	Ubuntu 10.04 x64
Ubuntu	14098	Ubuntu 10.04 x32
Ubuntu	345791	Ubuntu 13.04 x32
Ubuntu	350076	Ubuntu 13.04 x64
Ubuntu	433240	Ubuntu 12.10 x32
Ubuntu	459444	LAMP on Ubuntu 12.04
Ubuntu	473123	Ubuntu 12.10 x64
Ubuntu	473136	Ubuntu 12.10 x64 Desktop
Ubuntu	962304	Ubuntu 13.10 x32
Ubuntu	1061995	Wordpress on Ubuntu 12.10
Ubuntu	1420643	MEAN on Ubuntu 12.04.3
Ubuntu	1505447	Ubuntu 12.04.3 x64
Ubuntu	1505527	Ubuntu 12.04.3 x32
Ubuntu	1505699	Ubuntu 13.10 x64
Ubuntu	1608711	Ruby on Rails on Ubuntu 12.10 (Nginx + Unicorn)
Ubuntu	1687372	Redmine on Ubuntu 12.04
Ubuntu	1860934	Ghost 0.4.0 on Ubuntu 12.04

## 3.7 Command Line Interface

Doto provides a CLI tool for creating/destroying/managing resources on Digital Ocean from the command line. Many arguments can be prepended with an optional `-wait`, which instructs doto not to return control of the prompt until an event has completed. This is especially useful during creation and destruction of droplets.

### 3.7.1 Droplets

Example:

```
$ doto start --name Random --size_id 66 --image_id 2158507 --region_id 1 --ssh_key_ids 89221
```

List of arguments:

- start
- listdroplets
- power-on
- power-off
- rebuilt
- terminate
- info

### 3.7.2 SSH Keys

Example:

```
$ doto createkey -o file_name
```

List of arguments:

- createkey
- deletekey
- listkeys

### 3.7.3 Images

Example:

```
$ doto image "image name" show/destroy
```

List of arguments:

- image
- listimages

### 3.7.4 Snapshot

Example:

```
$ doto snapshot "droplet name" "snapshot name"
```

List of arguments:

- snapshot

## 3.8 Requirements

- requests ( $\geq 2.0.1$ )
- pycrypto ( $\geq 2.6.1$ )
- six ( $\geq 1.3.0$ )

## 3.9 Release Notes

- New CLI tool





---

## Indices and tables

---

- *genindex*
- *modindex*
- *search*